

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Дипломна робота

бакалавра

**з теми: “ ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ
АДМІНІСТРАТИВНОЇ МОДЕЛІ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ НА
БАЗІ ТЕХНОЛОГІЧНОГО СТЕКУ MEAN ”**

Виконав: студент 3 курсу, KNms1-B21 групи
спеціальності 122 “Комп'ютерні науки”

Мозолюк Микола Ігорович

Керівник:

Понеділок Вадим Віталійович

старший викладач кафедри комп'ютерних
наук, кандидат технічних наук

Рецензент:

Зеленський О. В.

доцент кафедри математики,

кандидат фізико-математичних наук, доцент

Кам'янець-Подільський, 2024 р.

АНОТАЦІЯ³

ВСТУП⁴

**РОЗДІЛ 1 ОЗНАЙОМЛЕННЯ З ВИМОГАМИ ДО
АДМІНІСТРАТИВНОЇ ПАНЕЛІ**⁶

1.1 Постановка задачі⁶

*1.1.1 Характеристика об'єкта автоматизації*⁶

*1.1.2 Вимоги до програми*⁷

1.2 Аналіз підходів для створення адміністративної панелі⁷

1.3 Актуальність виконання проекту¹⁰

1.4 Визначення структури вхідних даних¹²

1.5 Визначення структури вихідних даних¹³

1.6 База даних¹³

Висновки до розділу 1¹⁴

РОЗДІЛ 2 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ПАНЕЛІ¹⁶

2.1 Технічні характеристики комп'ютера та зовнішніх пристроїв¹⁶

2.3 Огляд стеку MEAN¹⁸

Висновки до розділу 2¹⁹

РОЗДІЛ 3 РОЗРОБКА АДМІНІСТРАТИВНОЇ ПАНЕЛІ²¹

3.1 Інтерфейс програми²¹

3.2 Програмування та тестування²⁵

*3.2.1 Технології проекту*²⁵

*3.2.2 Огляд функціональності*³⁰

*3.2.3 Етапи налагодження та тестування*³¹

*3.2.4 Тестування*³²

Висновки до розділу 3³⁶

ВИСНОВКИ³⁸

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ³⁹

ДОДАТОК А⁴⁰

ДОДАТОК Б⁴²

АНОТАЦІЯ

У даній роботі представлено теоретичні й практичні аспекти розробки адміністративної панелі для веб-додатків з використанням сучасних технологій фронтенду і бекенду. Робота включає аналіз вимог до адміністративної панелі, розгляд підходів для її створення, а також розробку інтерфейсу та функціональності за допомогою стеку MEAN (MongoDB, Express.js, Angular, Node.js).

Метою роботи є створення ефективної адміністративної панелі, яка дозволяє керувати інтернет-магазином, включаючи управління товарами, замовленнями та користувачами, забезпечуючи зручний інтерфейс та надійну взаємодію з базою даних.

Під час розробки використовувалися такі технології: Angular для фронтенду, Node.js та Express.js для бекенду, MongoDB для зберігання даних, а також Angular Material та Bootstrap для створення користувацького інтерфейсу та забезпечення адаптивності.

Результати роботи показують, що використання цих технологій забезпечує високу якість та функціональність адміністративної панелі, роблячи процес керування веб-додатком інтуїтивно зрозумілим та ефективним.

Ключові слова: веб-додаток, адміністративна панель, Angular, MEAN, Node.js, MongoDB, інтернет-магазин, управління товарами, управління замовленнями, адаптивний інтерфейс.

ВСТУП

Інтернет-магазини стають все більш важливими майданчиками для бізнес-транзакцій в сучасному цифровому середовищі, і вони також потребують ефективних управлінських рішень для своїх адміністративних операцій. Наша робота полягає у створенні та впровадженні адміністративної панелі з використанням технологічного стеку MEAN (Express.js, Angular, Node.js, MongoDB) для інтернет-магазину.

Сучасні вимоги до систем адміністрування стають все більш актуальними через поширення та складність інтернет-магазинів. З метою створення надійних та ефективних інструментів управління, в цій дипломній роботі буде розглянуто та оцінено можливості фреймворків Express.js, Angular, Node та MongoDB з точки зору їх використання в стеку MEAN.

Актуальність теми

Успіх інтернет-магазинів все більше залежить від їхньої здатності ефективно управляти своїми операціями, оскільки індустрія електронної комерції стає все більш конкурентною і розвиваються нові технології. Для онлайн-підприємств стратегічно важливою стає побудова адміністративної моделі, яка спрощує процеси та пропонує інтуїтивно зрозумілий інтерфейс для адміністраторів.

Мета роботи

Основною метою цієї дипломної роботи є дослідження та оцінка фреймворків MongoDB, Express.js, Angular та Node.js з точки зору їх застосування в стеку MEAN для створення адміністративних інтерфейсів інтернет-магазинів. Особливі завдання включають в себе:

1. знання специфікацій, необхідних для адміністративної панелі інтернет-магазину.
2. порівняння та аналіз технологій для створення адміністративних інтерфейсів
3. створення та розгортання адміністративної панелі з використанням фреймворків Express.js, Angular, Node.js та MongoDB.

4. оцінка надійності та ефективності створених інструментів.

Об'єкт та предмет дослідження

Об'єктом дослідження є адміністративна модель інтернет-магазину, а предметом - процес створення та реалізації адміністративних інтерфейсів на основі фреймворків Node.js, Angular або AngularJS, Express.js та MongoDB.

Метою даного дослідження є розробка та побудова зручних та ефективних інтерфейсів для адміністраторів, що дозволить покращити управління інтернет-магазинами за рахунок використання сучасних технологій.

Практичне значення

Кваліфікація бакалавра має практичне значення для створення ефективної адміністративної панелі для інтернет-магазину, яка полегшує управління та підвищує продуктивність адміністраторів. Надійність, масштабованість і швидкість розробки, які забезпечує технологічний стек MEAN (MongoDB, Express.js, Angular і Node.js), є важливими для конкурентоспроможності в сучасній електронній комерції. Розроблена панель є універсальним інструментом для бізнесу, і вона може служити основою для подальших удосконалень і адаптацій відповідно до потреб різних онлайн-магазинів.

Апробація результатів

Результати досліджень були оприлюднені на науковій конференції здобувачів вищої освіти за підсумками науково-дослідної роботи у 2023-2024 навчальному році, яка відбулася 9-10 квітня 2024 року в м. Кам'янець-Подільський.

Структура роботи

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1 ОЗНАЙОМЛЕННЯ З ВИМОГАМИ ДО АДМІНІСТРАТИВНОЇ ПАНЕЛІ

Дуже важливим в розробці є правильна постановка задачі, узгодження всіх нюансів, а також складання чіткого технічного завдання для чіткого розуміння завдання та його оцінці.

1.1 Постановка задачі

Важливим інструментом для управління кожною частиною вашого інтернет-магазину є адміністративна панель. Вона повинна забезпечувати високий рівень безпеки, зручності та надійності. Визначення цілей і основних обов'язків панелі є частиною виконаної роботи.

1.1.1 Характеристика об'єкта автоматизації

Об'єкт автоматизації – адміністративна панель інтернет-магазину “ОК-MARKET”, включає в себе такі підсистеми:

- Управління товарами: дозволяє адміністраторам додавати нові товари, редагувати інформацію про існуючі товари, завантажувати excel таблицю з товарами, видаляти товари з каталогу.
- Управління категоріями: дозволяє адміністраторам додавати нові категорії, редагувати інформацію про існуючі категорії, видаляти категорії.
- Управління замовленнями: включає перегляд та обробку замовлень, зміну їх статусів, та сповіщення клієнтів про стан їхніх замовлень.
- Управління відгуками: Перегляд відгуків, та відповіді на них.
- Відображення статистики сайту.

1.1.2 Вимоги до програми

Програма повинна відповідати наступним вимогам:

- Функціональні специфікації:

1. Управління товарами включає додавання, зміну та видалення позицій.
2. Управління категоріями включає додавання, зміну та видалення категорії або ж категорії з усіма наслідковими категоріями.
3. Управління замовленнями: перевірка, обробка та зміна статусу замовлень.
4. Управління відгуками: перевірка, та можливість відповіді.
5. Аналітика та звіти: аналіз даних та створення звітів.

- Вимоги, які не є функціональними:

1. Простота у використанні: нові користувачі можуть швидко ознайомитися з інтуїтивно зрозумілим інтерфейсом.
2. Безпека: захист приватної інформації від сторонніх очей, шифрування даних і захист від вторгнень, таких як SQL-ін'єкції та XSS.
3. Продуктивність: швидкий час відгуку та ефективна обробка запитів.
4. Масштабованість: здатність системи рости і легко включати нові функції в майбутньому.

1.2 Аналіз підходів для створення адміністративної панелі

Вибір оптимального технологічного стеку для цієї процедури може мати великий вплив на масштабованість, ефективність та продуктивність створеного рішення. Для оцінки придатності технологічного стеку MEAN для розробки та розгортання адміністративної панелі інтернет-магазину розглянемо кожен його елемент (MongoDB, Express.js, Angular, Node.js).

1. MongoDB [8] - це база даних документів, яка робить роботу з даними гнучкою та простою. MongoDB дозволяє обробляти та зберігати велику кількість даних в адміністративній панелі інтернет-магазину, включаючи інформацію про замовлення, клієнтів, товари тощо.

2. Node.js та Express.js [7, 1] - Для створення веб-додатків використовується фреймворк Node.js під назвою Express.js. Express.js полегшує розробку серверної частини адмінпанелі, пришвидшуючи створення маршрутів, обробників запитів та інших важливих завдань. З іншого боку, можливість створювати масштабовані та ефективні серверні JavaScript-додатки за допомогою Node.js дозволяє розробникам працювати над проектом швидше та ефективніше.

3. Angular [6] - Потужний фреймворк для розробки веб-додатків на TypeScript називається Angular. За допомогою Angular ви можете розробляти складні та динамічні інтерфейси в адміністративній панелі вашого інтернет-магазину. Він також пропонує розширені функції, такі як управління замовленнями, фільтрація товарів, звітність тощо.

Порівняльна таблиця фреймворків для розробки адміністративних панелей:

Фреймворк	Мова програмування	Складність вивчення	Гнучкість інтерфейсу	Розширені функціональні можливості
Angular	TypeScript	Висока	Складні, потужні	Фільтрація, звітність, управління
React	JavaScript	Середня	Прості, динамічні	Базові функціональності
Vue.js	JavaScript	Низька	Прості, гнучкі	Середні функціональність

Таблиця 1.1. Порівняльна таблиця фреймворків

Таблиця 1.1 відображає переваги та недоліки кожного фреймворку:

Angular:

Переваги:

- Потужний фреймворк з великими можливостями для створення складних інтерфейсів та функціональності.
- Зручна інтеграція з Google сервісами та Firebase.

Недоліки:

- Висока складність вивчення, що може ускладнити початковий процес розробки.

React:

Переваги:

- Швидкий та динамічний розвиток завдяки великій екосистемі та підтримці від Facebook.
- Простий у використанні та вивченні, особливо для досвідчених користувачів JavaScript.

Недоліки:

- Базовий функціонал, який може вимагати великої кількості доповнень для складних проектів.

Vue.js:

Переваги:

- Гнучкий та легкий у використанні фреймворк, що підходить як для початківців, так і для досвідчених розробників.
- Легка інтеграція з іншими бібліотеками та сервісами.

Недоліки:

- Середня функціональність, яка може бути недостатньою для деяких складних проектів.

Тому Angular є найкращим варіантом з кількох причин:

1. Angular - це надійний фреймворк, який пропонує чудовий потенціал для створення складних і корисних інтерфейсів. Він також має розширену

функціональність. Це дозволяє створити адміністративну панель з багатьма функціями, які можуть знадобитися інтернет-магазину, наприклад, управління замовленнями, фільтрація та звітність.

2. Проста інтеграція сервісів: Angular легко інтегрується з сервісами Firebase та Google. Це означає, що ви можете використовувати Firebase для створення таких функцій, як автентифікація користувачів та управління базами даних, а також з легкістю використовувати інструменти та сервіси Google, такі як Google Analytics, Карти, Диск тощо.

3. Для Angular існує велика та активна спільнота розробників, яка пропонує допомогу, часті оновлення та різноманітні розширення. Це означає, що ви можете використовувати розширення, щоб покращити свій проект і знайти швидкі відповіді на свої запитання та труднощі.

4. Таким чином, потужний набір функцій Angular, легка взаємодія з іншими сервісами та активна спільнота розробників, яка сприяє зростанню та підтримці проектів, роблять його мудрим вибором для розробки адмінпанелі інтернет-магазину

1.3 Актуальність виконання проекту

У сучасному бізнес-середовищі ефективне управління інтернет-магазином є життєво важливим для досягнення конкурентних переваг і задоволення потреб клієнтів. Інтернет-магазини стають все більш популярними завдяки простоті використання, широкому асортименту товарів і можливості купувати товари без відвідування фізичних магазинів. Зростання кількості онлайн-магазинів вимагає створення простих і ефективних адміністративних панелей, які дозволяють керувати різними елементами електронної комерції.

Розробка адміністративної панелі для інтернет-магазину на базі технологічного стека MEAN [4] (MongoDB, Express.js, Angular.js, Node.js) є важливим проектом, оскільки:

1. Стек MEAN дуже популярний серед розробників завдяки своїй гнучкості, можливостям створення масштабованих додатків і можливості використовувати одну мову програмування (JavaScript) на всіх рівнях розробки. Це зменшує витрати на навчання та підтримку коду та гарантує швидкий процес розробки.

2. Компоненти стеку MEAN дозволяють створювати додатки, які є надзвичайно продуктивними та масштабованими. Angular дозволяє створювати динамічні та інтерактивні інтерфейси користувача, NoSQL база даних MongoDB дозволяє гнучко зберігати дані, а Express.js і Node.js забезпечують швидку обробку запитів.

3. Інтернет-магазини обробляють велику кількість даних, включаючи інформацію про товари, замовлення, користувачів та їхню поведінку. Для безперебійної роботи магазину та прийняття розумних бізнес-рішень ефективно управління та обробка цих даних є критично важливими.

4. Рутинні завдання, такі як обробка замовлень, управління товарами, аналіз продажів і взаємодія з клієнтами, можна автоматизувати, щоб зменшити людські помилки, підвищити продуктивність і знизити витрати.

5. Адміністративна панель, яка зручна та функціональна, дозволяє швидко реагувати на зміни на ринку, адаптуватися до потреб клієнтів і надавати чудовий обслуговування. Це підвищує задоволеність споживачів і лояльність до бренду.

Таким чином, розробка адміністративної панелі для інтернет-магазину на базі стеку MEAN є надзвичайно важливим завданням, яке сприяє підвищенню продуктивності бізнесу, надає конкурентні переваги та задовольняє потреби сучасних користувачів.

1.4 Визначення структури вхідних даних

Вхідні дані для адміністративної панелі включають:

- Дані про товари:
 1. Категорія
 2. Назва товару
 3. Код
 4. Ціна
 5. Кількість
 6. Знижка
 7. Початок знижки
 8. Кінець знижки
 9. Опис
 10. Відео
 11. Фотографії
 12. Бренд товару
 13. Країна виробник
 14. Вага
 15. Розмір в упаковці
 16. Розмір без упаковки

- Дані про замовлення:
 1. Ідентифікатор замовлення
 2. Дата замовлення
 3. Статус замовлення
 4. Список товарів у замовленні

- Дані про категорію:
 1. Назва категорії
 2. Фото категорії
 3. ID батьківської категорії

1.5 Визначення структури вихідних даних

Вихідні дані включають:

- Звіти про продажі:
- Загальна кількість продажів за певний період
- Дохід
- Найбільш популярні товари
- Аналітичні дані:
- Тенденції продажів
- Поведінка користувачів
- Ефективність маркетингових кампаній
- Повідомлення для користувачів:
- Інформація про зміни статусу замовлення
- Сповідення про нові товари чи знижки

1.6 База даних

MongoDB - це високопродуктивна документно-орієнтована система управління базами даних NoSQL, яка пропонує гнучке зберігання даних. Її було обрано через наступні переваги:

- зміни в структурі даних можуть бути внесені з легкістю і без необхідності перепроєктування всієї бази даних завдяки гнучкій схемі.
- горизонтальне масштабування дозволяє ефективно управляти великими обсягами даних шляхом додавання нових серверів.
- підтримка складного пошуку та агрегації: дозволяє виконувати складні операції з даними безпосередньо в базі даних.

Заснована у 2007 році, компанія 10gen (зараз MongoDB Inc.) розробила MongoDB. Її основна мета - запропонувати масштабований та адаптований фреймворк для роботи з великими обсягами даних. Багато компаній по всьому

світу використовують MongoDB для різноманітних ініціатив, від простих веб-додатків до складних аналітичних систем.

За допомогою функцій MongoDB ви можете виконувати деякі дії безпосередньо на сервері бази даних. Для написання цих функцій можна використовувати JavaScript, що надає більше можливостей для кастомізації та маніпулювання даними.

Таким чином, використання бази даних MongoDB даних забезпечує ефективну обробку та зберігання інформації, що є необхідною умовою для нормальної роботи адміністративної панелі інтернет-магазину.

Висновки до розділу 1

Розділ 1 «Ознайомлення з вимогами до адміністративної панелі» зосереджується на тому, наскільки важливо правильно поставити завдання, узгодити деталі та скласти чітке технічне завдання для успішної розробки адміністративної панелі. Основні висновки, зроблені в цьому розділі:

- адміністративна панель забезпечує безпеку, зручність і надійність інтернет-магазину «ОК-MARKET» і є важливим інструментом для управління ним.
- основні підсистеми панелі, включаючи управління товарами, категоріями, замовленнями, відгуками та статистикою, забезпечують ефективне управління процесами магазину.
- програма повинна бути простою у використанні, безпечною для даних, високопродуктивною та масштабованою.
- завдяки гнучкості, продуктивності та можливості використовувати JavaScript на всіх рівнях розробки технологічний стек MEAN (MongoDB, Express.js, Angular, Node.js) є ідеальним вибором.
- створення ефективної адміністративної панелі має вирішальне значення для конкурентоспроможності та задоволення потреб клієнтів.
- чітка структура вхідних і вихідних даних полегшує обробку та зберігання даних.

- MongoDB ефективно зберігає та обробляє дані завдяки масштабованості, гнучкій схемі та підтримці складного пошуку.

Таким чином, правильна постановка задачі, визначення вимог і структур даних, а також використання сучасних технологій забезпечують успішну розробку та впровадження адміністративної панелі для ефективного управління інтернет-магазином.

РОЗДІЛ 2 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ПАНЕЛІ

2.1 Технічні характеристики комп'ютера та зовнішніх пристроїв

- Процесор - Intel Core i7-6500U з графікою Intel HD Graphics 520 (2,5 ГГц, до 3,1 ГГц с технологией Intel Turbo Boost, кеш 4 МБ, 2 ядра)
- Intel Core i5-6200U з графікою Intel HD Graphics 520 (2,3 ГГц, до 2,8 ГГц с технологией Intel Turbo Boost, кэш 3 МБ, 2 ядра)
- Intel Core i3-5005U з графікою Intel HD Graphics 5500 (2,0 ГГц, кеш 3 МБ, 2 ядра)
- Intel Pentium N3710 з графікою HD Graphics 405 (1,60 ГГц, до 2,56 ГГц з технологією Intel Turbo Boost, кеш 2 МБ, 4 ядра)
- Intel Celeron N3060 з графікою Intel HD Graphics 400 (1,6 ГГц, до 2,48 ГГц з технологією Intel Turbo Boost, кеш 2 МБ, 2 ядра)
- Набір мікросхем інтегрований з процесором
- Відеокарта - Intel HD 520
- Антибліковий 15,6-дюймовий WLED HD екран з тонкою рамкою (1366 x 768) SVA1
- Порт підтримує дозвіл до 2560 x 1200 при 60 Гц
- Пам'ять - DDR4-2133 (швидкості передачі аж до 2133 МТ / с)
- Один роз'єм SODIMM
- Підтримка двоканальної пам'яті. Клієнт не має доступу до пам'яті в т. ч. для її розширення
- Макс .: До 16 ГБ

2.2 Вибір програмних засобів та операційної системи

UNIX-подібні операційні системи, побудовані на ядрі Linux, зазвичай називають ОС Linux. Це один з найвідоміших прикладів розробки вільного програмного забезпечення з відкритим кодом. На відміну від пропрієтарних операційних систем, таких як Microsoft Windows і MacOS X, їхній вихідний

код є вільно доступним для будь-кого, хто може використовувати, модифікувати і розповсюджувати його без жодних витрат.

Переваги:

- виконувати кілька завдань одночасно. Оскільки процеси в Linux не залежать один від одного, кожному з них відводиться певна кількість часу.
- Linux підтримує багато користувачів. Це дозволяє дуже великій кількості людей працювати одночасно.
- Linux підтримує багато різних файлових систем, як застарілих, так і сучасних.
- Обмін програмним забезпеченням. У пам'ять завантажується лише одна копія виконуваного коду програми, якщо кілька користувачів одночасно виконують на комп'ютері одне й те саме завдання або якщо один користувач запускає одну й ту саму роботу кілька разів.
- частина жорсткого диска, що знаходиться на спеціальному розділі підкачки, використовується для роботи програм, які потребують багато пам'яті. Завдяки такому дизайну можлива робота з обмеженим обсягом оперативної пам'яті, і частина пам'яті може бути вивільнена на жорсткий диск.
- використовуйте графічний робочий процес без оболонки. Хоча графічна оболонка Linux не є обов'язковою, вона є важливим компонентом. Ефективніше використовувати термінал, а не графічний інтерфейс користувача.
- відсутність вірусів.
- різні дистрибутиви
- стабільність роботи - важливий фактор для кожного користувача при виборі операційної системи. Кожна версія, що випускається, проходить ретельне тестування як спільнотою, так і розробниками. Виправлення помилок і збоїв відбувається майже миттєво, а оновлення стають доступними звичайним користувачам, тільки якщо вони задовольняють усім вимогам стабільності.

2.3 Огляд стеку MEAN

До складу стеку MEAN входять наступні елементи:

- MongoDB - гнучка NoSQL база даних, яка зберігає дані в документах, що нагадують JSON. Вона орієнтована на документи. MongoDB - чудовий варіант, якщо ви працюєте з великою кількістю даних, які швидко змінюються. Підтримується висока продуктивність і горизонтальне масштабування.

- Express.js - це гнучкий і легкий веб-фреймворк Node.js, який полегшує створення онлайн-додатків і API. Express.js - це простий фреймворк, який дозволяє легко керувати часом виконання, маршрутизацією та обробкою запитів під час розробки серверних додатків.

- Angular від Google - це потужний фронтенд-фреймворк для створення динамічних односторінкових додатків (SPA). Angular пропонує високу продуктивність, модульну архітектуру та двостороннє прив'язування даних. Це дозволяє програмістам ефективно керувати станом додатку та розробляти складні користувацькі інтерфейси.

- Node.js - це серверна платформа, заснована на JavaScript-движку V8 від Google. За допомогою Node.js ви можете використовувати неблокуючу стратегію вводу/виводу для створення серверних додатків, які є одночасно масштабованими та високопродуктивними. Розробка та підтримка додатків спрощується завдяки тому, що Node.js використовує одну і ту ж мову програмування для клієнтської та серверної частин.

Переваги використання стеку MEAN:

- оскільки JavaScript є основою всього стеку, розробники можуть використовувати єдину мову як для серверної, так і для клієнтської частини, що спрощує створення та підтримку додатків.

- кожен компонент стеку MEAN пропонує високий ступінь гнучкості та можливість масштабування додатків у відповідь на зростання бізнес-вимог.

- функціональність розробляється набагато швидше завдяки готовим рішенням і бібліотекам MEAN.

- для кожного компонента стеку існує значна спільнота розробників, яка пропонує допомогу, часті оновлення та вдосконалення інструментів.

Використання стека MEAN:

- MongoDB пропонує швидке та гнучке зберігання даних для користувачів, замовлень та інформації про товари.

- Express.js дозволяє легко створювати RESTful API, які полегшують взаємодію між клієнтськими та серверними компонентами програми.

- Angular надає адміністраторам швидкий і простий у використанні інтерфейс, який дозволяє ефективно керувати онлайн-бізнесом.

- Node.js забезпечує відмінну швидкість роботи на стороні сервера, що дозволяє ефективно управляти системними ресурсами та швидко обробляти запити.

Висновки до розділу 2

Розділ 2 «Огляд технологій для розробки адміністративної панелі» обговорює технічні характеристики комп'ютерів, програмні засоби та стек MEAN, необхідні для створення адміністративної панелі. Основні висновки, зроблені в цьому розділі:

- Технічні характеристики обладнання: представлено кілька моделей процесорів, починаючи від Intel Core i7-6500U і закінчуючи Intel Celeron N3060, з описом відповідних характеристик, таких як частота, кеш і кількість ядер.

- Відеокарта Intel HD 520 та антибліковий екран роблять роботу приємною.

- Пам'ять DDR4-2133, яка підтримує до 16 ГБ, гарантує оптимальну продуктивність системи.

Вибір операційної системи та програмного забезпечення:

- Linux є найкращим вибором через відкритий код, стабільність, багатозадачність, багатокористувацьку роботу, різноманітні файлові системи та відсутність вірусів.

- Linux пропонує широкий вибір дистрибутивів, дозволяє працювати через термінал і ефективно використовує оперативну пам'ять.

Огляд стеку MEAN:

- MongoDB — це NoSQL-база даних, яка дозволяє обробляти та зберігати дані гнучко.

- Express.js — веб-фреймворк Node.js, який полегшує створення API та управління логікою сервера.

- Angular: фронтенд-фреймворк для динамічних односторінкових додатків з високою продуктивністю, розроблений Google.

- Node.js — це платформа на базі JavaScript, яка використовується для неблокуючої обробки запитів і високої продуктивності.

Переваги використання стеку MEAN:

- Масштабованість і гнучкість для задоволення потреб компанії.
- Швидкість розробки завдяки бібліотекам і готовим рішенням.
- Активна спільнота розробників, яка постійно підтримує інструменти та надає регулярні оновлення.

Використання стеку MEAN:

- MongoDB дозволяє швидко зберігати інформацію.
- Express.js робить створення RESTful API простішим.
- Angular пропонує зручний інтерфейс для адміністративної панелі.
- Node.js дозволяє керувати ресурсами та обробляти серверні запити швидко.

Отже, правильний вибір програмного забезпечення та апаратного забезпечення, а також використання стеку MEAN гарантують, що розробка та функціонування адміністративної панелі для інтернет-магазину будуть ефективними.

РОЗДІЛ 3 РОЗРОБКА АДМІНІСТРАТИВНОЇ ПАНЕЛІ

3.1 Інтерфейс програми

Бібліотеки Angular Material і Bootstrap були використані для створення інтерфейсу адміністративної панелі. З використанням цих технологій було створено зручний, естетично привабливий і функціональний інтерфейс користувача.

Angular Material — це набір компонентів інтерфейсу користувача, розроблених спеціально для Angular додатків. Ці компоненти відповідають рекомендаціям дизайну Material Design, який був розроблений Google. Основні переваги використання Angular Material:

- компоненти Angular Material відповідають рекомендаціям Material Design, що забезпечує єдиний стиль і користувацький досвід у всьому додатку.
- Angular Material пропонує широкий набір готових компонентів, таких як кнопки, форми, діалогові вікна, меню, таблиці тощо, що значно прискорює процес розробки.
- компоненти автоматично адаптуються до різних розмірів екранів, що забезпечує зручне користування як на настільних комп'ютерах, так і на мобільних пристроях.
- Angular Material розроблено спеціально для Angular, щоб забезпечити гладке використання всіх можливостей фреймворку та забезпечити гладку інтеграцію.

Bootstrap - це популярний фреймворк для розробки інтерфейсів користувача, який пропонує широкий спектр інструментів, щоб допомогти вам створити дизайн, який відповідає вашим потребам. Найважливіші переваги використання шаблону:

- шаблон дозволяє створювати веб-додатки, які можуть автоматично адаптуватися до різних розмірів екранів і пристроїв.
- Bootstrap пропонує великий вибір готових стилів і компонентів, таких як сітка, форми, кнопки, навігаційні панелі та багато іншого, що дозволяє швидко створювати інтерфейси.

- компоненти Bootstrap оптимізовані для роботи в різних браузерах, щоб додаток працював у всіх популярних веб-браузерах.

- шаблон має простий і зрозумілий синтаксис, що дозволяє легко та швидко створювати функціональні та стильні інтерфейси.

Причини вибору кутового матеріалу та шаблону:

- використання готових компонентів і стилів значно скоротить час, необхідний для розробки та тестування інтерфейсу.

- обидві бібліотеки дотримуються сучасних тенденцій і рекомендацій у дизайні.

- Material and Bootstrap компоненти дозволяють створювати інтерфейси, які виглядають однаково добре на різних пристроях і екранах.

- обидві технології мають активну спільноту розробників, яка надає доступ до великої кількості ресурсів, прикладів і документації.

Завдяки використанню Angular Material та Bootstrap адміністративна панель отримала сучасний, зручний і функціональний інтерфейс, який відповідає вимогам користувачів і забезпечує ефективне управління інтернет-магазином.

The screenshot shows the 'OK-MARKET' admin interface. On the left is a navigation menu with 'Товари' (Products) selected. The main area displays a table of products with the following data:

Код товару	Назва	Ціна	Кількість	Дії
DKNY-110023	Годинник наручний жіночий DKNY NY2769 золотий	5800	2	Редагувати / Видалити
PA-132121	Гумовий ущільнювач Profile W PARKSIDE коричневий	300	4	Редагувати / Видалити
PF-330014	Гумовий ущільнювач Profile W PARKSIDE білий	300	14	Редагувати / Видалити
ZOO-330002	Іграшка для собак стейк зоофагі	700	0	Редагувати / Видалити
VT-111213	Джемпер чоловічий з гудзиками LIVERGY M (48/50) Темно-сірий	550	1	Редагувати / Видалити
US-330006	Органайзер-тримач для багажника Ultimate SPEED Чорний-Червоний	550	1	Редагувати / Видалити

At the bottom right of the table, there is a pagination control showing 'Items per page: 10' and '1 - 10 of 116'.

Рисунок 3.1.1. Головне меню сторінки товарів

На рисунку 3.1.1 ми можемо побачити сторінку з товарами, на ній виведено основну інформацію по товарах, є можливість створити новий товар, загрузити excel таблицю з товарами, редагувати або ж видалити товар. Також бачимо реалізовану пагінацію за допомогою Angular Paginator на сторінці. Усі кнопки і елементи зроблені за допомогою Angular Material [11].

Рисунок 3.1.2. Сторінка створення товару

Рисунок 3.1.3. Сторінка редагування товару

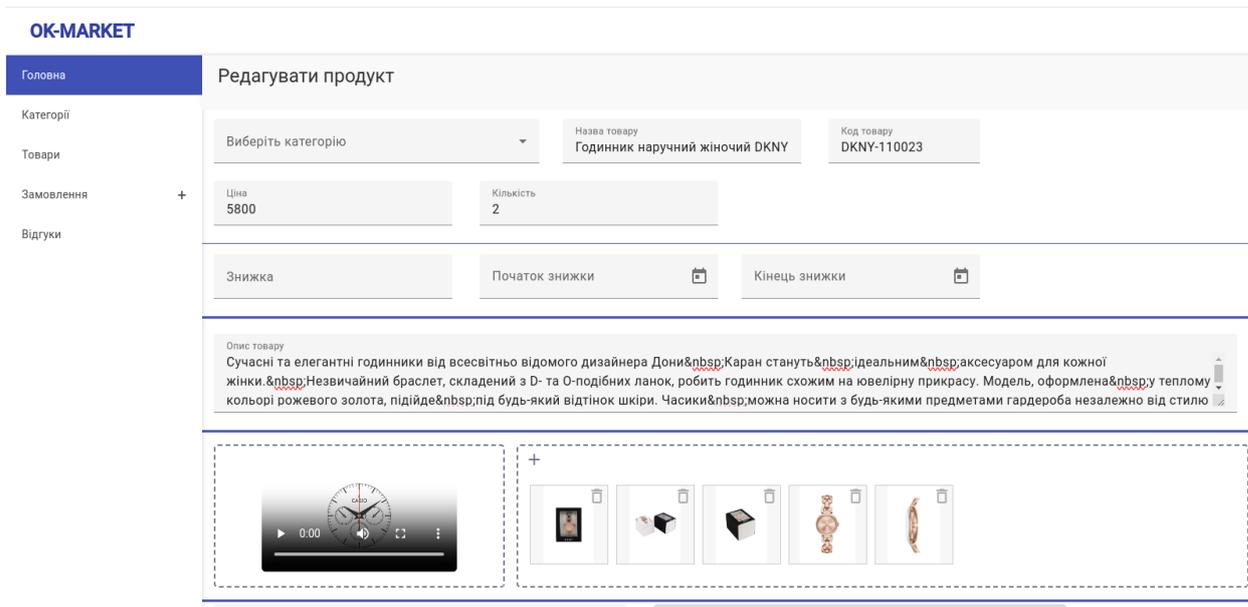


Рисунок 3.1.4. Сторінка редагування товару

На рисунках 3.1.2, 3.1.3, 3.1.4 представлено сторінку створення/редагування товару, Усі елементи на цій сторінці створені за допомогою бібліотека Angular Material, також додавання фото та відео до продукту було реалізовано за допомогою бібліотеки NGX-Dropzone.

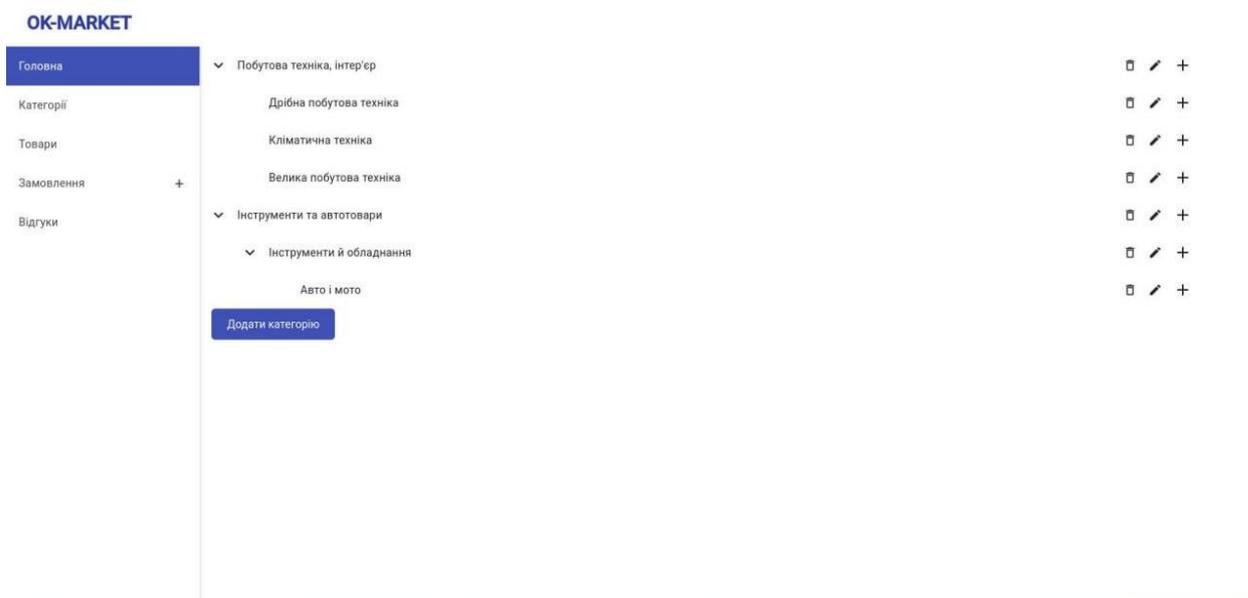


Рисунок 3.1.5. Головне меню сторінки категорій

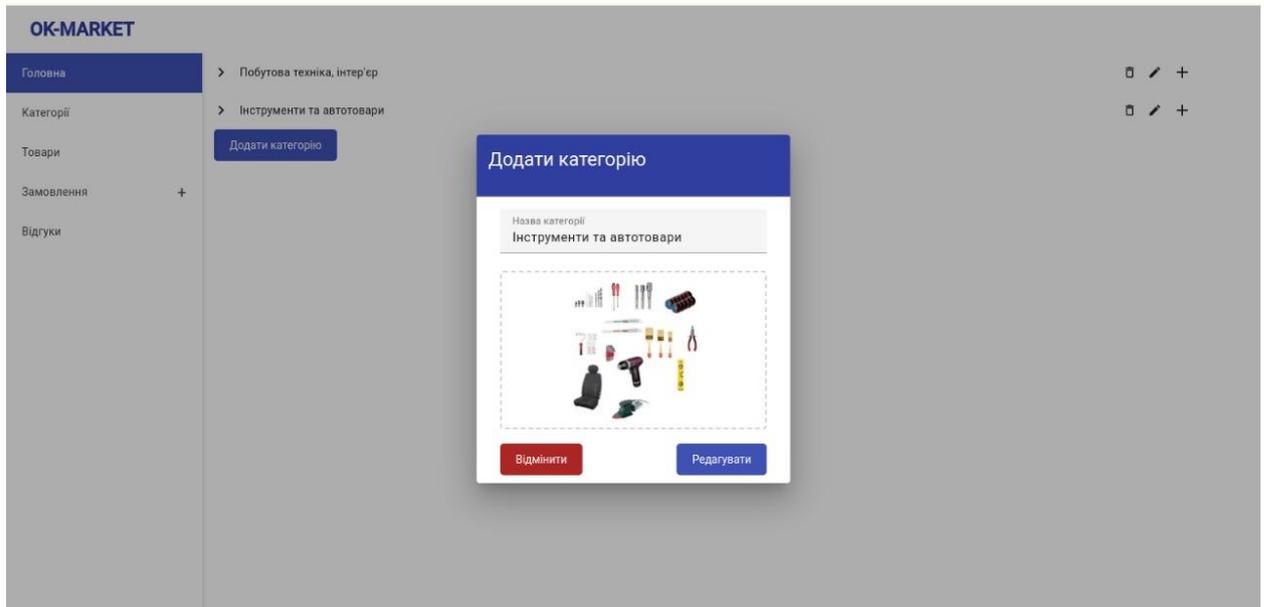


Рисунок 3.1.6. Сторінка редагування категорій

На рисунках 3.1.5 та 3.1.6 відображено сторінку управління категоріями, тут ми можемо створювати, видаляти та редагувати категорії.

3.2 Програмування та тестування

Програмування та тестування є останнім і чи не найважливішим етапом моєї роботи, адже від правильності та ефективності реалізації коду залежить стабільність та надійність всього проекту. В цьому розділі я детально розглянув використані залежності для фронтенд та бекенд частин проекту, а також опишу процес програмування та тестування.

3.2.1 Технології проекту

Стек MEAN включає Angular для фронтенду та Node.js для бекенду для створення адміністративної панелі інтернет-магазину. У цьому розділі буде детально розглянуто використані залежності для обох частин проекту, а також розглянуто процес програмування та тестування.

Фронтенд:

Angular - популярний фреймворк для створення веб-додатків - використовується для реалізації фронтенд-частини адміністративної панелі.

Ось список головних залежностей фронтенду проекту:

```
{
  "name": "ok-admin",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve --live-reload",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^15.2.7",
    "@angular/cdk": "^15.2.6",
    "@angular/common": "^15.2.0",
    "@angular/compiler": "^15.2.0",
    "@angular/core": "^15.2.0",
    "@angular/forms": "^15.2.0",
    "@angular/material": "^15.2.6",
    "@angular/platform-browser": "^15.2.7",
    "@angular/platform-browser-dynamic": "^15.2.0",
    "@angular/router": "^15.2.0",
    "angular-polyfills": "^1.0.1",
    "bootstrap": "^5.2.3",
    "core-js": "^3.30.2",
    "ng2-dragula": "^3.2.0",
```

```

"ng2-file-upload": "^3.0.0",
"ngx-drag-drop": "^15.1.0",
"ngx-dropzone": "^3.1.0",
"ngx-owl-carousel": "^2.0.7",
"rxjs": "~7.8.0",
"tslib": "^2.3.0",
"zone.js": "~0.12.0"
},
"devDependencies": {
"@angular-devkit/build-angular": "^15.2.5",
"@angular/cli": "~15.2.5",
"@angular/compiler-cli": "^15.2.0",
"@types/jasmine": "~4.3.0",
"jasmine-core": "~4.5.0",
"karma": "~6.4.0",
"karma-chrome-launcher": "~3.1.0",
"karma-coverage": "~2.2.0",
"karma-jasmine": "~5.1.0",
"karma-jasmine-html-reporter": "~2.0.0",
"typescript": "~4.9.4"
}
}

```

Основні залежності:

- **@angular/animations**, **@angular/cdk**, **@angular/common**,
@angular/compiler, **@angular/core**, **@angular/forms**, **@angular/material**,
@angular/platform-browser, **@angular/platform-browser-dynamic**,
@angular/router: Основні пакети Angular для роботи з компонентами,
формами, анімаціями та маршрутизацією.

- **angular-polyfills**: Додаткові полімери для підтримки різних браузерів.
- **bootstrap**: CSS-фреймворк для швидкого створення адаптивних інтерфейсів.
- **core-js**: Бібліотека для використання сучасних JavaScript-функцій у старих браузерах.
- **ng2-dragula, ng2-file-upload, ngx-drag-drop, ngx-dropzone, ngx-owl-carousel**: Бібліотеки для інтерактивних елементів інтерфейсу, таких як перетягування, завантаження файлів та каруселі.
- **rxjs, tslib, zone.js**: Бібліотеки для реактивного програмування та допоміжні бібліотеки для Angular.

Основні devDependencies:

- **@angular-devkit/build-angular, @angular/cli, @angular/compiler-cli**: Інструменти для побудови та компіляції Angular додатків.
- **@types/jasmine, jasmine-core, karma, karma-chrome-launcher, karma-coverage, karma-jasmine, karma-jasmine-html-reporter**: Бібліотеки та інструменти для модульного та інтеграційного тестування.
- **typescript**: Мова програмування, яка є основою для Angular додатків.

Бекенд:

Бекенд частина адміністративної панелі реалізована за допомогою Node.js та Express.js. Ось список основних залежностей проекту для бекенду:

```
{
  "name": "ok-admin",
  "version": "0.0.0",
  "scripts": {
    "start": "node server.js",
    "dev": "nodemon server.js"
```

```
},  
"private": true,  
"dependencies": {  
  "express": "^4.18.2",  
  "mongoose": "^6.9.0",  
  "body-parser": "^1.20.1",  
  "cors": "^2.8.5",  
  "jsonwebtoken": "^8.5.1",  
  "bcryptjs": "^2.4.3"  
},  
"devDependencies": {  
  "nodemon": "^2.0.20",  
  "eslint": "^8.30.0",  
  "jest": "^29.3.1"  
}  
}
```

Основні залежності:

- **express**: Легкий та гнучкий фреймворк для створення веб-додатків на Node.js.
- **mongoose** [9]: Бібліотека для роботи з MongoDB, яка забезпечує зручну модель даних.
- **body-parser**: Мідлвер для обробки JSON даних у запитах.
- **cors**: Мідлвер для налаштування політики доступу з різних доменів.
- **jsonwebtoken**: Бібліотека для створення та валідації JSON Web Token (JWT), які використовуються для аутентифікації.
- **bcryptjs**: Бібліотека для хешування паролів.

Основні devDependencies:

- **nodemon**: Інструмент для автоматичного перезапуску сервера при зміні коду.
- **eslint**: Інструмент для забезпечення якості коду та дотримання стандартів.
- **jest**: Фреймворк для тестування JavaScript коду.

3.2.2 Огляд функціональності

Архітектура додатку:

- **Фронтенд**: Інтерфейс користувача виконаний на Angular. Він пропонує двостороннє зв'язування даних та динамічне оновлення сторінок без перезавантаження. Основні елементи інтерфейсу наступні:

1. **Dashboard**: основна адміністративна панель, що містить зведення важливих показників.
2. **Управління товарами** - модуль, який дозволяє додавати, змінювати та видаляти товари.
3. **Перегляд і контроль замовлень** можливий за допомогою модуля «Замовлення».

- **Бекенд**: Бекенд додатку побудований з використанням Node.js та Express.js. Основними компонентами бекенду є наступні:

1. **Реалізація RESTful API** дозволяє користувачам спілкуватися з фронтендом. За допомогою API можна отримати доступ до бази даних для виконання завдань CRUD (створення, читання, оновлення, видалення).
2. **Модулі безпеки**: JSON Web Tokens (JWT) використовуються для реалізації автентифікації та авторизації користувачів.

- **База даних**: Дані про товари, замовлення та користувачів зберігаються в MongoDB. База даних організована наступним чином:

1. **Колекція продуктів**: відстежує дані про продукти.

2. Колекція замовлень: зберігає дані, пов'язані із замовленнями.

3. Колекція категорій: відстежує дані про категорії

4. Колекція відгуків: відстежує дані про відгуки

Виконання основних вимог:

- Авторизація та аутентифікація. Для того, щоб отримати доступ до адміністративної панелі, користувачі повинні спочатку пройти автентифікацію. Доступ до певних модулів можна отримати за допомогою авторизації на основі ролі користувача.

- Управління продуктами - це застосування функцій CRUD, таких як ціноутворення, контроль кількості і завантаження зображень для продуктів.

- Керування замовленнями. Перегляд, редагування та зміна статусу замовлення.

- Керування категоріями. Перегляд, редагування та зміна категорій.

- Керування відгуками. Перегляд, можливість відповіді на відгук.

3.2.3 Етапи налагодження та тестування

Вже виявлені програмні помилки знаходять і виправляють під час унікального етапу розробки програми, який називається налагодженням програми. Але перш ніж виправити помилку, її потрібно знайти. Тестування - це процедура, яка передбачає запуск програми для пошуку та виправлення помилок.

Фундаментальні ідеї організації тестування

Виправлення помилки вимагає, по-перше, повного виявлення її причин, які часто не є очевидними одразу. По-друге, ця дія має негативний психологічний вплив, оскільки програмісту доводиться визнавати, що саме його помилка є причиною непрацездатності програми. Крім того, налагодження - це процес, який просто призупиняється, поки тестування знаходить додаткові помилки.

Якщо програма не працює або видає неточні результати зі стратегічної точки зору потрібне тестування. Існує два основні підходи до налагодження програм, кожен з яких має унікальні характеристики:

- при розробці програми використовувати всі методи, які дозволять мінімізувати кількість дефектів, що містяться в ній. Якщо помилки виявлено, основним методом їх усунення є ручна перевірка та аналіз тексту програми.

- використання комп'ютерного програмного забезпечення, або так званих відладчиків, для пошуку помилок є рекомендованим підходом. Різновидом цього методу є трасування - функція певних програмних середовищ, яка дозволяє запускати програму крок за кроком і стежити за значеннями змінних під час виконання програми. Оскільки відладчик програми може працювати точніше, ніж компілятор, наприклад, встановлюючи невизначені змінні в нуль або по-іншому розподіляючи пам'ять, цей метод не гарантує, що всі помилки будуть виявлені.

3.2.4 Тестування

Тестування є важливим етапом розробки програмного забезпечення, який забезпечує його надійність та якість. В цьому розділі описані основні методи тестування, які були використані при розробці адміністративної панелі для інтернет-магазину на базі стеку MEAN.

Модульне тестування

Модульне тестування - це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яка може бути протестованою. У процедурному програмуванні модулем вважають окрему функцію або процедуру.

Модульне тестування коду є невід'ємною частиною життєвого циклу розробки програмного забезпечення. Модульні тести також формують основу для проведення регресійного тестування, тобто вони гарантують, що система

буде вести себе відповідно до сценарію, коли збільшаться нові функціональні можливості або зміняться існуючі.

Для тестування компонентів адміністративної панелі було використано Jasmine [15].

Jasmine для Angular:

- **Test Bed Angular Test Bed (АТВ)** використовується для створення тестового середовища, яке імітує реальне оточення додатку. Це дозволяє проводити тести компонентів, сервісів та модулів Angular.
- **Component Testing** це перевірка правильності роботи окремих компонентів, включаючи перевірку шаблонів, зв'язування даних та подій.
- **Service Testing** для тестування сервісів, які забезпечують обробку логіки додатку та взаємодію з API.

Інтеграційне тестування

Інтеграційне тестування перевіряє взаємодію між різними модулями системи. Воно спрямоване на виявлення проблем, які можуть виникати при інтеграції окремих компонентів у єдину систему.

Для тестування кінцевих точок API використовується бібліотека **Supertest** [14]. Supertest — це бібліотека для тестування HTTP-запитів, яка дозволяє перевіряти коректність роботи RESTful API.

Основні концепції Supertest:

- **Test Fixture:** виконується підготовка, необхідна для виконання тестів, такі як запуск серверу та підключення до бази даних.
- **Test Case:** перевірка окремих кінцевих точок API, включаючи GET, POST, PUT та DELETE запити.
- **Test Suite:** об'єднання кількох тестових випадків для комплексного тестування API.
- **Test Runner:** запуск тестів та збір результатів, відображення статусу виконання тестів.

Системне тестування

Системне тестування охоплює повне тестування всієї системи для виявлення можливих дефектів та проблем. Воно включає тестування всіх функцій адміністративної панелі в реальних умовах експлуатації.

Основні етапи тестування [2]:

1. Етап перший. Аналіз вимог

Умови і критерії роботи системи, як правило, визначаються замовником або менеджером проєкту в процесі спілкування з клієнтом, або аналізом стандартів та нормативної документації. Ці вимоги можуть бути як функціональними, так і нефункціональними.

Тестувальник працює над статичним тестуванням вимог: досліджує їх повноту, логічність, однозначність і визначеність, також знаходить слабкі місця в тестовому покритті і виявляє можливі ризики. Виникають обставини, коли наявні вимоги не тестуються, але використовуються на етапі дизайну і розробки. І вже готовий продукт направляється у відділ QA. Тоді кількість знайдених дефектів, вартість та масштаб їх виправлення можуть значно зрости.

2. Етап другий. Процес дизайну

Один з найважливіших складових етапів, тому що користувачеві доведеться постійно взаємодіяти з системою. Продукт повинен бути інтуїтивно зрозумілий у використанні, мати приємний інтерфейс, а також бути оптимізованим (користувач робить мінімальні дії для досягнення результату).

На даному етапі тестувальник перевіряє існуючі прототипи ПЗ на відповідність вимогам замовника, коректності відображення візуальних елементів і зручності використання.

3. Етап третій. Розробка

Під час процесу розробки системи необхідно провести модульне, інтеграційне та системне тестування. Для початку тестуються окремі компоненти програми. Це дозволяє виявити, в якій саме частині коду є помилка і досить швидко усунути її. Ще модульне тестування допомагає

відстежити, чи не призвела зміна коду до появи нових помилок в уже перевічених місцях продукту. Також це допомагає краще зрозуміти роль кожного модуля системи. Усі знайдені дефекти, як правило, виправляються в коді без формального їх опису (внесення звітів в баг-трекер).

Інтеграційне тестування дозволяє перевірити, як компоненти коду об'єднуються і взаємодіють один з одним. Також воно визначає зв'язки між модулями, обладнанням або різними системами.

І, нарешті, переходимо до системного тестування. На цій стадії мається на увазі перевірка всіх компонентів і модулів в цілому. Це необхідно для зменшення ймовірності дефектів, пов'язаних з особливостями поведінки системи в різних середовищах. В ході цих перевірок виявляються такі баги, як: нераціональне використання ресурсів системи, несумісність з оточенням, збій або неправильне функціонування функціоналу та інше.

4. Етап четвертий. Процес тестування та дебагінгу

Дебагінг (налагодження) – це процес, під час якого знаходять і усувають помилки програми.

На даному етапі тестувальники перевіряють систему на наявність дефектів незалежно від того, чи відбувалось це раніше. Проводиться повне тестування інтерфейсу та функціоналу продукту. Всі виявлені помилки повинні бути задокументовані в баг-трекері. Також слід провести регресійне тестування, щоб дослідити систему на предмет дефектів, які могли з'явитися після усунення інших багів.

По завершенню процесу налагодження потрібно надати оцінку якості продукту, наскільки відповідають вимоги замовника реальній роботі системи.

5. Етап п'ятий. Експлуатація та підтримка

Після того, як продукт надходить у реліз, залишається необхідність у тестуванні, так як буде відбуватися оновлення ПЗ, будуть з'являтися нові баги, які були випущені або баги, пов'язані з експлуатацією кінцевого користувача. В такому випадку потрібне втручання відділу QA.

У разі виявлення користувачами тих чи інших пост-релізних багів, інформація про них передається у вигляді звітів про помилки команді розробки, яка, в залежності від серйозності проблеми, або негайно випускає виправлення (т.зв. hot-fix), або відкладає його до наступної версії програми.

Всі зміни, що вносяться до програмного забезпечення, необхідно ретельно протестувати. ПЗ повинне продовжувати виконувати спочатку закладені в ньому бізнес-функції і не порушувати працездатність інших функцій і всієї системи в цілому.

Процес тестування ПЗ охоплює всі етапи життєвого циклу розробки. Він безперервний, тривалий і вимагає наявності досить досвідченої команди тестувальників, щоб охопити всі етапи тестування. А моніторинг та аналіз всього процесу допомагає спланувати і, в разі необхідності, внести зміни для підвищення ефективності подальших завдань. Ця частина сучасного процесу розробки ПЗ допомагає замовнику, команді розробників, і головне, кінцевому користувачеві отримати продукт високої якості.

Завдяки всебічному тестуванню адміністративної панелі забезпечується висока якість та стабільність роботи системи, що відповідає сучасним вимогам бізнесу та користувачів.

Висновки до розділу 3

У цьому розділі розроблено адміністративну панель для інтернет-магазину з використанням Angular Material і Bootstrap. У процесі розробки було отримано глибоке розуміння основних концепцій і потенціалу технологій. Створення сучасного та зручного інтерфейсу користувача було легко завдяки широкому вибору готових компонентів і стилів Angular Material. Bootstrap дозволяє швидко адаптувати дизайн і підтримувати широкий діапазон пристроїв і розмірів екранів.

У цій адміністративній панелі присутні всі функції, необхідні для ефективного управління інтернет-магазином. Він містить модулі для керування користувачами, товарами, замовленнями та звітами, що робить його

потужним інструментом для власників і керівників магазинів. Щоб зробити адміністративну панель зручною та задовольнити користувачів, вона була розроблена відповідно до сучасних стандартів дизайну та UX.

Таким чином, розробка адміністративної панелі для інтернет-магазину з використанням Angular Material і Bootstrap виявилася досить виправданою, коли йдеться про надання якісного та функціонального інтерфейсу користувача.

ВИСНОВКИ

У цій кваліфікаційній роботі було вивчено та розглянуто сучасні технології та методи розробки веб-додатків, які використовують Angular, Angular Material і Bootstrap. Розуміння принципів SPA (односторінкових додатків), архітектурних підходів і методів реалізації з використанням Angular Framework значно покращилося.

Angular ідеально підходить для розробки великих і складних веб-додатків завдяки своїй масштабованості, модульності та гнучкості. Також Angular Material і Bootstrap надають широкий вибір готових компонентів і стилів, що значно спрощує процес створення сучасного та адаптивного дизайну інтерфейсу користувача.

У процесі розробки було створено інтернет-магазин з використанням цих технологій, а саме адміністративної панелі, яка дозволяє користувачам ефективно керувати користувачами, товарами та замовленнями. Результати показали, що використання Angular, Angular Material і Bootstrap дозволяє швидко створювати функціональні та естетично привабливі додатки, які відповідають сучасним стандартам і вимогам користувачів.

Основні результати даної роботи підкреслюють, наскільки важливо мати глибоке розуміння технологій, знати, як вибрати найкращі інструменти для кожного етапу розробки та гарантувати, що кінцевий продукт буде високої якості. Крім того, наголошується на тому, що знання та навички повинні постійно розвиватися, щоб успішно працювати в сучасному інформаційному середовищі.

Отже, ця робота демонструє важливість і переваги використання Angular, Angular Material та Bootstrap у розробці веб-додатків і надає підґрунтя для подальших досліджень і вдосконалення в цій галузі. Також розроблена адміністративна панель відповідає всім вимогам та виконує всі заплановані функції поставленим на етапі планування задачі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Express JS - JavaScript web-фреймворк для програмної платформи Node.js. URL: <https://brander.ua/technologies/expressjs> (дата звернення: 04.06.2024)
2. ЕТАПИ ТЕСТУВАННЯ ПЗ. URL: <https://training.qatestlab.com/blog/technical-articles/software-testing-stages/> (дата звернення: 09.06.2024)
3. Top 10 Most Popular JavaScript Frameworks to Choose in 2024. URL: <https://medium.com/@evincedevelop/top-10-most-popular-javascript-frameworks-to-choose-in-2024-269453cdaf35> (дата звернення: 07.06.2024)
4. MEAN Stack Development in 2024: What's New and What to Expect: <https://medium.com/@mobiloittetech060/mean-stack-development-in-2024-whats-new-and-what-to-expect-895eccce3780> (дата звернення: 05.06.2024)
5. Introduction to the Angular docs. URL: <https://v17.angular.io/docs> (дата звернення: 04.06.2024)
6. Node.js Documentation. Node.js Official Documentation. URL: <https://nodejs.org/>. (дата звернення: 04.06.2024)
7. MongoDB Documentation URL: <https://www.mongodb.com/docs/> (дата звернення: 05.06.2024)
8. Mongoose. URL: <https://mongoosejs.com/> (дата звернення: 05.06.2024)
9. Full-Stack vs MEAN Stack vs MERN Stack: The Right Technology Stack for You in 2024. URL: <https://radixweb.com/blog/full-stack-vs-mean-stack-vs-mern-stack-development> (дата звернення: 04.06.2024)
10. Angular Material URL: <https://material.angular.io/> (дата звернення: 05.06.2024)
11. NPM URL: <https://www.npmjs.com/> (дата звернення: 08.06.2024)
12. Vue.js: <https://vuejs.org/> (дата звернення: 08.06.2024)
13. Supertest. URL: <https://www.npmjs.com/package/supertest> (дата звернення: 06.06.2024)
14. Jasmine . URL: <https://jasmine.github.io/> (дата звернення: 06.06.2024)
15. MongoDB: The Definitive Guide (3rd Edition). URL: <https://www.oreilly.com/library/view/mongodb-the-definitive/9781491954459/> (дата звернення: 07.06.2024)
16. Building APIs with Express. URL: <https://www.pluralsight.com/courses/building-apis-express> (дата звернення: 03.06.2024)
17. Testing Node.js Applications. URL: <https://www.oreilly.com/library/view/testing-nodejs-applications/9781492042216/> (дата звернення: 07.06.2024)

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ АДМІНІСТРАТИВНОЇ ПАНЕЛІ НА ОСНОВІ MEAN СТЕКУ

ВСТУП

Це технічне завдання описує розробку адміністративної панелі для керування даними в веб-додатку з використанням MEAN стеку (MongoDB, Express, Angular, Node.js). Адміністративна панель буде використовуватись для управління інформацією про користувачів, контент та системні налаштування інтернет магазину. Проєкт розробляється в рамках бакалаврської роботи.

ПІДСТАВИ ДЛЯ РОЗРОБКИ

Розробка адміністративної панелі здійснюється в рамках виконання бакалаврської роботи з метою покращення керування інтернет магазином та забезпечення зручності адміністрування.

ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою розробки є створення функціональної адміністративної панелі, яка забезпечить зручний інтерфейс для управління даними в системі, а також покращення ефективності роботи адміністраторів.

ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Вимоги до функціональних характеристик:

Програмний засіб повинен бути простим у використанні та інтуїтивно зрозумілим для адміністратора. В ньому мають бути реалізовані такі функції:

- Управління товарами: дозволяє адміністраторам додавати нові товари, редагувати інформацію про існуючі товари, завантажувати ексел таблицю з товарами, видаляти товари з каталогу.
- Управління категоріями: дозволяє адміністраторам додавати нові категорії, редагувати інформацію про існуючі категорії, видаляти категорії.
- Управління замовленнями: включає перегляд та обробку замовлень, зміну їх статусів, та сповіщення клієнтів про стан їхніх замовлень.
- Управління відгуками: Перегляд відгуків, та відповіді на них.

- Відображення статистики сайту.

Вимоги до надійності:

1. Безвідмовність: Система повинна працювати без збоїв протягом тривалого періоду часу під нормальними умовами використання.
2. Відмовостійкість: У разі виникнення помилок або збоїв, система повинна відновлюватися до робочого стану без втрати інформації.
3. Захищеність: Система повинна мати механізми захисту від навмисних або ненавмисних дій, які можуть призвести до збоїв або втрати даних.
4. Масштабованість: Система повинна підтримувати збільшення кількості користувачів та даних без зниження продуктивності та якості обслуговування.
5. Безпека даних: Всі дані, що зберігаються або передаються, повинні бути захищені від несанкціонованого доступу та зміни.

Умови експлуатації:

Програмний засіб може використовуватись на комп'ютерах та ноутбуках з сучасними веб-браузерами.

ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу документації входять бакалаврська робота та технічне завдання.

СТАДІЇ І ЕТАПИ РОЗРОБКИ

- Початкова підготовка: Дослідження та аналіз необхідних для реалізації процесів.
- Визначення вимог до розроблюваної системи.
- Проектування та створення програмного продукту на основі технічного завдання.
- Впровадження системи, тестування в реальних умовах експлуатації та внесення необхідних коригувань, виявлених під час дослідної експлуатації.

ДОДАТОК Б

ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ АДМІНІСТРАТИВНОЇ ПАНЕЛІ

Адміністративна панель інтернет-магазину "OK-MARKET" розроблена для ефективного управління всіма аспектами функціонування магазину. Вона повинна забезпечувати високий рівень безпеки, зручності та надійності, а також надавати інструменти для управління товарами, категоріями, замовленнями, відгуками та статистикою сайту. Визначення цілей і основних обов'язків панелі є частиною виконаної роботи.

Адміністративна панель інтернет-магазину "OK-MARKET" включає в себе наступні підсистеми:

- Управління товарами:
 1. Додавання нових товарів.
 2. Редагування інформації про існуючі товари.
 3. Завантаження Excel таблиці з товарами.
 4. Видалення товарів з каталогу.
- Управління категоріями:
 1. Додавання нових категорій.
 2. Редагування інформації про існуючі категорії.
 3. Видалення категорій.
 4. Управління замовленнями:
- Перегляд та обробка замовлень.
 1. Зміна статусів замовлень.
 2. Сповіщення клієнтів про стан їхніх замовлень.
- Управління відгуками:
 1. Перегляд відгуків.
 2. Відповіді на відгуки.
- Відображення статистики сайту:
 1. Перегляд аналітики та статистичних даних.